

Appendix

A Axioms of the System

There are three sets of axioms.

Specific axioms for features and sorts :

Let τ, τ' denote any sorts, and f denote any feature.

$$\begin{aligned} \forall x, y, z \quad x \stackrel{f}{\vdash} y \wedge x \stackrel{f}{\vdash} z \supset y = z \\ \forall x \quad \neg(x : \tau \wedge x : \tau') \text{ if } \tau \neq \tau' \\ \forall x, y \quad x : \tau \wedge y : \tau \supset x = y \text{ if } \tau \text{ is a value sort} \\ \forall x, y \quad \neg(x : \tau \wedge x \stackrel{f}{\vdash} y) \text{ if } \tau \text{ is a value sort} \end{aligned}$$

Congruence axioms for equality :

Let p denote any built-in predicate. The traditional congruence axioms are:

$$\begin{aligned} \forall x \quad x = x \\ \forall x, y \quad x = y \supset y = x \\ \forall x, y, z \quad x = y \wedge y = z \supset x = z \\ \forall x, y \quad x : \tau \wedge x = y \supset y : \tau \\ \forall x, y, z \quad x \stackrel{f}{\vdash} y \wedge x = z \supset z \stackrel{f}{\vdash} y \\ \forall x, y, z \quad x \stackrel{f}{\vdash} y \wedge y = z \supset x \stackrel{f}{\vdash} z \\ \forall \bar{x}, y \quad p(\bar{x}) \wedge x_i = y \supset p(\bar{y}) \end{aligned}$$

where i is some index in the list of variable \bar{x} and \bar{y} is identical to \bar{x} except that $y_i = y$.

Built-in predicate axioms :

They must not mention sorts and features. For example, disequality can be axiomatized by

$$\begin{aligned} \forall x, y \quad x \neq y \vee x = y \\ \forall x \quad \neg(x \neq x) \end{aligned}$$

Precedence constraints are axiomatized by

$$\begin{aligned} \forall x \quad \neg(x < x) \\ \forall x, y, z \quad x < y \wedge y < z \supset x < z \end{aligned}$$

The built-in predicates $>, \leq, \geq$ can then be defined from $<$ and equality.

B Constraint Satisfaction

B.1 The BFC case

We represent a BFC as a pair $\langle B \mid \Gamma \rangle$ where B is a built-in constraint and Γ an *unordered* list of sort and feature constraints (read conjunctively). \perp denotes the contradiction.

There are two sets of rewrite rules. The following rules correspond to simplifications of the BFCs.

$$\begin{aligned} \langle B \mid x \stackrel{f}{=} y, z \stackrel{f}{=} t, \Gamma \rangle &\mapsto \langle B \wedge y = t \mid x \stackrel{f}{=} y, \Gamma \rangle \text{ if } \vdash_T B \supset x = z \\ \langle B \mid x : \tau, y : \tau, \Gamma \rangle &\mapsto \langle B \mid x : \tau, \Gamma \rangle \text{ if } \vdash_T B \supset x = y \text{ and } \tau \text{ is not a value sort} \\ \langle B \mid x : \tau, y : \tau, \Gamma \rangle &\mapsto \langle B \wedge x = y \mid x : \tau, \Gamma \rangle \text{ if } \tau \text{ is a value sort} \end{aligned}$$

The following rules correspond to the detection of inconsistencies.

$$\begin{aligned} \langle B \mid \Gamma \rangle &\mapsto \perp \text{ if } \vdash_T \neg B \\ \langle B \mid x : \tau, y : \tau, \Gamma \rangle &\mapsto \perp \text{ if } \vdash_T B \supset x = y \text{ and } \tau \neq \tau' \\ \langle B \mid x : \tau, y \stackrel{f}{=} z, \Gamma \rangle &\mapsto \perp \text{ if } \vdash_T B \supset x = y \text{ and } \tau \text{ is a value sort} \end{aligned}$$

The following property justifies the algorithm

$$\langle B \mid \Gamma \rangle \mapsto \perp \text{ if and only if } \vdash_T \forall \neg (B \wedge \bigwedge_{c \in \Gamma} c)$$

B.2 The SFC case

We represent an SFC as an unordered list of BFCs prefixed with a sign (+ or -); by definition, one and only one component is positive. Let S be an SFC. The SFC-normal form of S is written S^* and is obtained by the following algorithm:

Let c_o be the BFC-normal form of the positive component of S .

If $c_o = \perp$ Then

Return \perp

Else

c_o is of the form $\langle B_o \mid \Gamma_o \rangle$

Let $\{\langle B_i \mid \Gamma_i \rangle\}_{i=1, \dots, n}$ be the list of negative components of S .

Foreach $i = 1, \dots, n$

Let c_i be the BFC normal form of $\langle B_o \wedge B_i \mid \Gamma_o, \Gamma_i \rangle$.

If there exists $i \in 1, \dots, n$ such that $c_i = \langle B \mid \Gamma \rangle$ and $\vdash_T B$ and Γ is empty Then

Return \perp

Else

Let $I = \{i \in 1, \dots, n \text{ such that } c_i \neq \perp\}$

Return $\{+c_o, \{-c_i\}_{i \in I}\}$

The following property justifies the algorithm

$$[+\langle B_o \mid \Gamma_o \rangle, \{-\langle B_i \mid \Gamma_i \rangle\}_{i=1}^n]^* = \perp \text{ if and only if } \vdash_T \forall \neg [(B_o \wedge \bigwedge_{c \in \Gamma_o} c) \wedge \bigwedge_{i=1}^n \neg (B_i \wedge \bigwedge_{c \in \Gamma_i} c)]$$